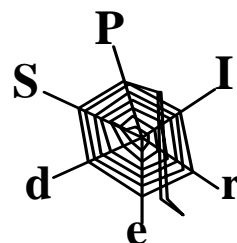


SPIDER— Koerier



Februari 2008 Nummer 1

www.st-SPIDER.nl

■ Redactioneel

Mocht u nog een mededeling, suggestie of een artikel hebben waarvan u denkt dat het interessant zou kunnen zijn voor de SPIDER leden, mail dan naar: koerier@st-SPIDER.nl.

■ Inhoudsopgave

■ Inhoudsopgave.....	1
■ Van het bestuur	1
■ LogicaCMG nieuwe sponsor van SPIDER!.....	2
■ Sponsoring Stichting SPIDER door LogicaCMG een feit!.....	3
■ De toegevoegde waarde van (Software) Configuration Management	5
■ Infotenties	9
■ Waarom je geen architect moet worden	10
■ System Architecture: The Silver Bullet?	14
■ What's Wrong With Agile Software Methods?	17
■ Call for papers 11e SPIDER Conferentie.....	22
■ CMMI roadmaps worden gepubliceerd bij het SEI	23
■ Laatste bijeenkomst van de metriecken werkgroep.....	23
■ Deelname in SPIDER	24
■ Nieuwsberichten & evenementenkalender	24
■ Colofon.....	24

■ Van het bestuur

De economie is wereldwijd volop in beweging. Omhoog en omlaag, pieken en dalen, het gaat allemaal steeds sneller. Gaat het nou goed, of toch niet? Zijn we uit het dal, richting een top, of staan we bij een volgende afgrond? Wie het weet mag het zeggen, en ja, er zijn er een hoop die het (denken te) weten, en het zeggen. Maar, snappen U en ik het nog?

Hetzelfde zie je binnen de ICT. Allerlei vernieuwingen in de recente jaren: Agile, lean, Six Sigma. Nieuwe regelgeving en standaards waaraan je als bedrijf moet voldoen.: Outsourcing, nearsourcing en insourcing, en allerlei andere vormen van multi site development en complexe samenwerkingsverbanden. En dan rijst de vraag: Hoe houden we de regie?

Dit is nu juist een gebied waar proces verbetering een oplossing biedt. Door methoden, technieken en best practices die organisaties helpen om veranderprocessen te managen en veranderingen daadwerkelijk doorgevoerd te krijgen en ook geborgd. Zaken als definiëren van de verandering, ondersteunen van het beslissen over de implementatie, controle houden over de operatie, inzicht krijgen en behouden in de resultaten zijn gesneden koek voor proces verbeteraars. En ze hebben daarbij ook nog inzicht in en ervaring met de menselijke factor in veranderprocessen.

Kat in het bakkie zou je dan denken. Toch niet helemaal. Jazeker, er zijn een hoop bedrijven die SPI specialisten betrekken bij bovenstaande veranderingen. En met succes, vaak ook als het niet direct om veranderingen gaan die te maken hebben met ICT of software ontwikkeling. Immers de vaardigheden van SPI mensen zijn over een veel breder terrein inzetbaar. Maar helaas, er zijn ook een hoop trajecten waarin SPI mensen (nog) niet betrokken worden. En dat terwijl ze er wel degelijk een toegevoegde waarde kunnen en willen bieden.

Ik denk dat er een aantal oorzaken hiervoor zijn. Allereerst leeft bij veel mensen nog steeds het beeld dat SPI specialisten (alleen) iets doen met modellen. Dus als het gaat over CMMI, EFQM/NKM, ISO audits, etc dan weet men ons wel te vinden, maar bij een verbetering van de Operational Excellence, of bij fusies, outsourcing of andere complexe veranderingen worden we vergeten. Daar kunnen we zelf wat aan doen. Namelijk laten

De activiteiten van SPIDER worden gesponsord door financiële bijdragen van:



Philips.com



Kza.nl



Sogeti.nl



LogicaCMG.com

zien dat dat iets is wat we ook kunnen, en al doen! We hebben de vaardigheden om dit te doen, we kunnen omgaan met complexe materie, hebben de kwaliteiten om met verschillende partijen te communiceren en bij elkaar te brengen. Laten we onze successen promoten (en ook leren van onze en andermans fouten) en onszelf op de kaart zetten.

Verder werken SPI specialisten vaak op de achtergrond, en zijn ze niet zo zichtbaar. Voor het resultaat is dat goed, immers we zorgen ervoor dat ontwikkelaars en managers het leren en daarna zelf kunnen, waarna we met een gerust hart weg gaan naar een volgende klus. Maar voor onszelf is dat funest, men ziet niet wat we gedaan hebben. En, een verandering lijkt vaak zo simpel, maar is het meestal niet. Als zo'n traject dan gecoached wordt door ervaren verandermanagers, die vooruit zien, problemen voorkomen, mensen helpen in de samenwerking en ze leren om met complexiteit om te gaan, dan gaat het als een zonnetje. Zie je wel, veranderen is helemaal niet moeilijk zegt men dan, en dat had best zonder de SPI specialist gekund. Niet dus! Moeten we het dan mis laten gaan, om op te vallen? Dat willen we niet (en diep in ons hart kunnen we dat ook niet), en vervolgens ziet dus niemand wat we achter de schermen allemaal gedaan hebben, en hoe we bijgedragen hebben aan het resultaat.

SPIder kan en wil een bijdrage leveren aan de zichtbaarheid van SPI en kwaliteit. Binnen SPIder zien we breed, daarbinnen vallen ook verandertrajecten die niet (alleen) met software te maken hebben. Veranderen is mensenwerk, en ook die menselijke factoren hebben de aandacht binnen SPIder, denk maar eens terug aan onze 10^e conferentie in 2007. En als het gaat om kwaliteit, daarbij kijken we ook naar de effecten op tijd en geld. Immers, minder fouten betekent ook minder verstoringen bij de ontwikkeling en vrijgave van producten en diensten, dus ook sneller en goedkoper.

Hoe gaan we in 2008 de zichtbaarheid van SPI en kwaliteit verhogen? We gaan dit jaar thema's programmeren die SPI in de breedte laten zien, d.w.z. competentie en discipline overstijgend. Zoals de plenaire sessie op 6 maart a.s. over multi-site ontwikkelen. Hierbij wordt vanuit verschillende invalshoeken ingegaan op productontwikkeling door virtuele teams, teams die wederzijds afhankelijk zijn en gemeenschappelijke doelen en verantwoordelijkheden hebben, maar ook gekenmerkt worden door culturele, economische en geografische verschillen. Verder zitten er plenaire sessies in de pen over verbeteren en het toepassen van requirements engineering en over toepassingen van test

improvement, bijeenkomsten waarin we laten zien wat de rol en toegevoegde waarde van SPI en kwaliteit is in deze aanverwante vakgebieden.

Voor de 11^e Spider conferentie gepland op 29 september kijken we naar alle veranderingen die spelen in organisaties, en hoe dat SPI en kwaliteit daaraan bijdragen. Verderop in deze Koerier de call for papers: Als je ervaring hebt met proces en organisatieverandering, transformaties en (her)inrichting van (IT) organisaties dan wel met kwaliteitstrajecten (en wie heeft dat niet?), stuur dan een paper in voor de conferentie!

Ook zoeken we nog meer dan vroeger de samenwerking op. Op 6 maart is dat met het platform outsourcing. Voor andere sessies werken we samen met onze zusterverenigingen. En ook in de Q Society slaan we de brug tussen industrie, ICT, onderwijs en netwerken. Samen hebben we een missie: Verbetering van ICT en IT bedrijven en instellingen in Nederland, en ervoor zorgen dat professionals zich kunnen ontwikkelen en hun ervaringen in en rondom het vakgebied met elkaar delen ten faveure van de BV Nederland.

Ik wens je veel plezier bij het lezen van deze Koerier, en tot ziens bij SPIder!

Namens het bestuur
Ben Linders
Voorzitter SPIder

■ LogicaCMG nieuwe sponsor van SPIder!

Sinds 1 januari 2008 is LogicaCMG een sponsor van de stichting SPIder. In het recente verleden is LogicaCMG jarenlang een trouwe sponsor van SPIder geweest. Velen van u kunt zich waarschijnlijk het blauwe logo van CMG nog goed herinneren. Na een aantal jaren van afwezigheid is LogicaCMG opnieuw van harte welkom! Met hun sponsoring erkennen en herkennen ze het vakgebied van SPI en kwaliteit, en steunen ze de missie van SPIder om dit in Nederland verder te verbeteren.

Ook voor de medewerkers van LogicaCMG is dit goed nieuws, met daarbij dat ze nu ook gebruik kunnen maken van alle donateursvoordelen van SPIder die de sponsoring hun biedt. Zoals korting op de jaarlijkse SPIder conferentie, toegang tot alle informatie van SPIder, korting op de online boekhandel van Pearson, en diverse voordelen die we middels samenwerking met onze zusterverenigingen aan kunnen bieden.

In een apart artikel in deze Koerier gaan Eltjo Poort en Rob van der Stap nader in op de hernieuwde sponsorschap van LogicaCMG. Duidelijk is dat SPI en kwaliteits strategisch belangrijk gezien worden, en dat sluit perfect aan bij de missie van SPIder. Ik ben blij met deze samenwerking, en we gaan daar een succes van maken!

Namens het bestuur
Ben Linders

■ Sponsoring Stichting SPIder door LogicaCMG een feit!

Wij zijn verheugd u te kunnen zeggen dan LogicaCMG de Stichting SPIder heeft uitgekozen als een van de speerpunten voor IT en de professionalisering van IT die door de Stichting SPIder en door LogicaCMG van harte worden ondersteund

LogicaCMG is een gerenommeerde internationale IT provider die al meer dan 40 jaar in de IT opereert. LogicaCMG is op vele markten actief.

LogicaCMG is een kwaliteitsbewuste organisatie, een bedrijf waarin iedereen dezelfde kansen krijgt om zichzelf en daarmee het bedrijf verder te ontplooiën. LogicaCMG is al jaren in staat gebleken haar medewerkers uitdagend werk aan te blijven bieden, zowel bij onze klanten als intern. Ook de klanttevredenheid over de producten en diensten van LogicaCMG staat al vele jaren op een hoog peil.

LogicaCMG onderkent de volgende Lines of Business:

- Consulting
- [Energy and utilities](#)
- [Enterprise services](#)
- [Financial services](#)
- **Global service delivery**
- [HR and payroll services](#)
- [Industry, distribution, transport](#)
- [Public sector](#)
- [Security practice \(UK\)](#)
- [Space and defence \(UK\)](#)
- [Telecoms and media](#)
- [Testing services](#)

De International Lines of Business (ILOB) van LogicaCMG overbruggen afstand, cultuur en taal. Zo is de Global Service Delivery organisatie (GSD) een internationaal opererende divisie waarbij BPO, outsourcing en offshoring voor tal van klanten in zowel het thuisland (in ons geval Nederland) als in India plaatsvindt. GSD opereert conform een zgn. blended model. De klant doet zijn werk, of dat nu Banking, Insurance of Telecom is, terwijl

LogicaCMG de IT verzorgt. Waar en hoe dit gebeurd is voor de klant transparant. Hiervoor worden software factories en test centers ingezet conform het LogicaCMG Result model.

LogicaCMG opereert op verschillende markten.



'Markets' zijn de markten zoals LogicaCMG die ziet. Ze komen overeen met onze marktdivisies: Financial Services, Energy & Utilities, Industry Distribution & Transport, Public Sector en Telecoms. Deze markten zijn op hun beurt onderverdeeld in submarkten, bijvoorbeeld 'Oil & Gas' onder Energy & Utilities. Met deze verfijning wordt snel duidelijk aan welk type organisaties LogicaCMG haar diensten levert.

LogicaCMG werkt aan meerdere **Business Themes**

Business Themes, ook wel Vertical Propositions genoemd, zijn onderwerpen die in een bepaalde markt belangrijk zijn voor LogicaCMG of die specifiek zijn voor die markt, de klant. Zie hieronder enkele aandachtsgebieden waarin LogicaCMG opereert.

Financial Services

- *Payments*
- *Risk & Compliance*

Energy & Utilities

- *Oil & Gas reserves*
- *Retail costs & retail products*

Public Sector

- *E-Health*
- *Citizen Self Service/Citizen Participation*

LogicaCMG onderkent vele Competences

Naast Verticals zijn er Horizontal Propositions, oftewel de Competences. Deze competences zijn niet marktspecifiek, maar worden in meerdere markten geleverd. Test Management en Application Development zijn goede voorbeelden. Beide zijn aan te bieden aan zowel een bank als aan een ministerie. De Competences beschrijven de inhoud van de LogicaCMG dienstverlening; wát we doen.



Zoals mag worden verwacht heeft LogicaCMG in Nederland sinds jaar en dag een eigen SPI community. Dit is een selecte groep van zeer ervaren, actieve consultants die werkzaam zijn op het uitdagende gebied van kwaliteit en procesverbetering. Gezien de breedte van hun werkterrein is de SPI community een virtuele organisatie. Collega's met uiteenlopende achtergrond, kennis, kunde en ervaring en werkzaam in verschillende sectoren – maar allemaal verbonden door het 'SPI virus' – werken hierin nauw samen aan verdere professionalisering. De SPI SG is de enige stuurgroep binnen LogicaCMG waarin alleen maar consultants (en geen managers) deelnemen en het ook nog voor het zeggen hebben ook!

Application Development

Application Development Environments (Java, Oracle, etc. ontwikkelstraten)

Software Process Improvement (SPI)

De LogicaCMG SPI competentie (Software Process Improvement) richt zich op het verbeteren van productontwikkeling met als doel het verbeteren van softwarekwaliteit en het verbeteren van proces en organisatiekwaliteit van bedrijven die software producten en diensten voortbrengen. De aandacht is hierbij gevestigd op drie peilers:

- Proces- en performanceverbetering
- Requirements Engineering
- Software architectuur

Juist de combinatie van deze drie peilers vormt de kracht van de LogicaCMG SPI competentie. Met deze competentie beoordeelt LogicaCMG de vaardigheid van een ontwikkelorganisatie, inventariseert oorzaken van problemen en helpt organisaties daadwerkelijk deze problemen op te lossen via advisering, begeleiding, training en coaching enerzijds en door het managen van transformatieprogramma's, professionaliseringstrajecten en complexe IT inrichtingsprojecten anderzijds.

ICT Management

- Business Service Management
- Governance of Outsourced IT

Technical Software Engineering

- *Product Quality Improvement (PQI)*
- *Embedded System Design*

Infrastructure, Security & Architecture

- *Risk management*
- *Infrastructure architecture*

Enterprise Resource Management

- *Business Intelligence*
- *Testing Services*

Consulting & Management

- *Management/Business Consultancy*
- *Project Management*

Emerging Technologies

- Machine2Machine (M2M)
- Radio Frequency Identification (RFID)

Innovatie, RFID in de trein, het lijkt simpel maar je moet er maar op komen!!! Voorbeeld van een innovatie project uitgevoerd door LogicaCMG.

Binnenkort gaan we een aantal aangepaste stoelzittingen in een dubbeldekker plaatsen. In deze zittingen zullen diverse soorten draadloze sensoren geplaatst worden waarmee we kunnen meten of een stoel al dan niet bezet is. Het doel van deze pilot is om de technische haalbaarheid van "vrije zitplaatsdetectie" te evalueren. In deze pilot zal de bezettingsinformatie draadloos naar het WAM (Wireless Asset Management) backoffice van LogicaCMG worden verstuurd. Via webpages wordt vervolgens realtime getoond welke zitplaatsen bezet zijn.

Deze informatie kan weer op een aantal manieren worden gebruikt, namelijk:

- ✓ In de trein: om de reizigers aan te duiden waar zitplaatsen beschikbaar zijn.
- ✓ Op de perrons: om meer inzicht te geven waar men moet instappen of dat het wachten op een volgende trein de moeite waard is.
- ✓ Realtime monitoring van treinbezetting voor optimalisatie of validatie van de dienstregeling (in het bijzonder de materieel inzet).
- ✓ Of bijvoorbeeld het efficiënter besturen van de airconditioning (CO² gehalte), afhankelijk van de bezetting.

Tot slot

Wij zien uit naar een prettige samenwerking en wensen SPIDER nog vele succesvolle jaren toe. SPIDER is al meer dan tien jaar als enige onafhankelijk opererende organisatie op SPI gebied toonaangevend in Nederland. Als het aan LogicaCMG ligt mag dit zo nog jaren door gaan.

Hebt u verder nog vragen over LogicaCMG of wilt u met ons in contact treden dan zien wij uw reactie graag tegemoet,

Eltjo Poort, *NL Lead Technical Architect*
(eltjo.poort@logiacmg.com)

Rob van der Stap, *NL Technical Director*
(rob.van.der.stap@logiacmg.com)

■ De toegevoegde waarde van (Software) Configuration Management

Inleiding

Met de opkomst van CMM(I) in het afgelopen decennium, is ook de belangstelling voor Configuration Management toegenomen. Configuration Management (CM) en in het bijzonder Software Configuration Management (SCM) mag zich de laatste jaren in toenemende populariteit verheugen.

Over het belang van het hebben van een software configuration management systeem (bijvoorbeeld ClearCase, CM Synergy of SubVersion) zijn ondertussen talrijke artikelen geschreven, en de meeste software ontwikkelorganisaties hebben ondertussen dan ook een SCM systeem in gebruik. Ook over het gebruik van deze systemen is al het nodige geschreven. Zeker met de toenemende complexiteit van zowel producten als projecten ontkomt men er niet aan om vooraf de strategie voor SCM te bepalen om zodoende de projecten optimaal te kunnen ondersteunen.

Echter, SCM kent ook nog andere aspecten welke tot op heden enigszins onderbelicht zijn gebleven. Denk hierbij bijvoorbeeld aan Configuration Status Accounting (CSA) en Baseline Auditing (BA). Binnen CMMI wordt het genoemd als onderdeel van CM, maar over concrete invulling hiervan is tot op heden nog niet veel te vinden.

In dit artikel zal aan de hand van enkele voorbeelden geschetst worden hoe er een basis gelegd kan worden voor CSA en BA, en hoe SCM hiermee van toegevoegde waarde kan zijn voor het project.

De grootste toegevoegde waarde is dat het project een betere controle krijgt over de status van de leveringen en de inhoud hiervan.

Een positief neveneffect hiervan is dat dit indirect kan leiden tot verbeteringen in de processen rondom de totstandkoming van deze leveringen.

Tot slot laten deze voorbeelden is ook te zien dat er een sterke relatie bestaat tussen metrieke/statistieke en auditing en accounting.

Definities

Allereerst: wat wordt er eigenlijk bedoeld met CSA en BA? Googelen op deze termen leveren ruwweg 2.000.000 resultaten op. Hieronder volgen een tweetal definities welke gangbaar en werkbaar zijn in de praktijk.

Met **Configuration Status Accounting** wordt bedoeld dat er gerapporteerd wordt over de indicatoren die de status bepalen van de items welke onder versiebeheer staan. Deze items worden in CM terminologie aangeduid met de term Configuration Item (CI). De indicatoren waarover gerapporteerd wordt staan vervolgens vernoemd in het Software Configuration Management Plan (SCMP).

Enkele voorbeelden van zulke indicatoren zijn:

- Feedback uit het bouwproces
- Volwassenheid van het product (maturity)
- Status van problem reports en change requests op de CI's
- Consistentie van de levering

In het eerder genoemde SCMP staat ook wat de waarden van de indicatoren dienen te zijn bij de diverse statusovergangen tijdens het ontwikkeltraject. Bij deze statusovergang worden de indicatoren getoetst door middel van een **Baseline Audit**¹. Hierbij kijk je of alle indicatoren voldoen aan de gestelde criteria en of daarmee vervolgens overgegaan kan worden naar de volgende status (bijvoorbeeld van "getest" naar "vrijgegeven").

Door nu op regelmatige basis CSA uit te voeren, kan er tijdens het project al een beeld gevormd worden of aan de criteria voor de statusovergang voldaan kan gaan worden. Ook kunnen eventuele afwijkingen ten opzichte van de geplande waarden zo tijdig gedetecteerd worden.

Hiermee onderscheidt zich dan ook de status van CSA en BA.

CSA is zuiver informatief/indicatief. Rapportage kan dan ook uitstekend via een website, welke regelmatig opgedateerd wordt.

¹ Naast Baseline Audit bestaat ook de Configuration Audit. In de context van dit artikel mogen deze als gelijkwaardig beschouwd worden.

Omdat BA gebruikt wordt voor een formele statusovergang, heeft een BA rapportage een formeel karakter. Indien één of meerdere indicatoren niet de gewenste eindwaarde heeft bereikt, zal de impact hiervan besproken moeten worden met de betrokken partijen, en zullen er correctieve acties gedefinieerd moeten worden. In het ergste geval kan dit zelfs leiden tot het niet toekennen van een nieuwe status.

Overigens dient opgemerkt te worden dat de keuze van indicatoren per project verschillend kan zijn. Het kiezen hiervan dient te gebeuren in een samenspel tussen de configuration manager en betrokken partijen. De betrokken partijen geven in dit samenspel aan welke indicatoren van belang zijn, wat de gewenste waarden zijn en hoe deze waarden te interpreteren. De SCMer kijkt daarnaast ook nog naar de meetbaarheid van de indicatoren.

Enkele uitgewerkte voorbeelden:

Tot zover de theorie. De kunst is nu deze theorie te vertalen naar de eigen werkomgeving. Onderstaand een viertal uitgewerkte praktijkvoorbeelden ter inspiratie. Deze voorbeelden laten zien hoe BA en CSA in praktijk toegepast kan worden zonder daarbij afhankelijk te zijn van de gebruikte configuration management tooling.

Analyseren bouwresultaten

Als eerste voorbeeld een analyse die door veel SCMer's / buildmanagers gedaan wordt: een analyse van de bouwresultaten.

Het analyseren en administreren van bouwfouten op subsysteem- of moduleniveau kan op termijn inzicht verschaffen in de kwaliteit van aanleveren. Indien een bepaalde module regelmatig bouwfouten veroorzaakt, kan dit duiden op hiaten in het ontwikkelproces voor deze module.

Denk hierbij aan oorzaken als:

- Slechte of geen code-review
- Ontwikkelaar test niet alvorens te leveren
- Ontwikkelaar bouwt niet alvorens te leveren
- Ontwikkelaar weet niet welke versie van de interfaces te gebruiken

Door dit tijdens het project al te monitoren, kan er tijdig bijgestuurd worden.

Een meer diepgaande analyse van de log-files van een bouwproces geeft doorgaans aanvullende informatie welke het projectverloop kan beïnvloeden. Vooral waarschuwingen over het gebruik van oude interfaces die in een volgende versie niet meer ondersteund worden, of bepaalde methoden/functies die in een volgende versie van de compiler niet meer ondersteund worden, kunnen onverwachte problemen opleveren indien er een update plaatsvindt van interfaces of de compiler.

In het SCMP kan bijvoorbeeld gesteld worden dat er bij een bepaalde mijlpaal geen oude interfaces meer gebruikt mogen worden. CSA kan in dit geval een indicatie geven of de mijlpaal gehaald gaat worden of niet, en eventueel hoeveel werk hier nog voor gedaan moet worden.

Als uit de BA blijkt dat er niet aan de afspraak voldaan wordt, zal een impact analyse uit moeten wijzen of dit wel of niet belemmering vormt voor het passeren van de mijlpaal.

Maturity Grid

Een andere meting welke goed te gebruiken is voor CSA en BA is de maturity grid (zie kader). In het SCMP wordt gedefinieerd welk maturity-niveau de software dient te hebben bij welke mijlpaal.

Hiermee wordt het controleren of dit niveau gehaald is een typische BA activiteit.

Echter, waarom wachten tot het naderen van de mijlpaal om deze controle uit te voeren?

Door deze meting al regelmatig gedurende het project uit te voeren (als onderdeel van CSA), kan er tijdig een voorspelling gedaan worden of het gewenste niveau op tijd gehaald kan worden, en daar waar nodig correctieve acties geïnitieerd worden.

Als er bijvoorbeeld nog 100 problemen opgelost moeten worden om het gewenste maturity-niveau te bereiken, maar de mijlpaal is al over twee weken, zal gekeken moeten worden of het oplossen van 50 problemen in een week realistisch is of niet.

Overigens blijkt op zo'n moment het nut van het bewaren van historische data. Op basis van deze data kan een realistische voorspelling gedaan worden over het aantal problemen dat per week opgelost kan worden.

Maturity grid: methode waarbij de volwassenheid van het project wordt gemeten aan de hand van het aantal ingediende problemen, de ernst van deze problemen (severity) en hun actuele status.

Weergave gebeurt via een matrix (grid). (L1-L2-L3 geeft de ernst van de problemen aan)

	L1	L2	L3
Nieuw			
In behandeling			
opgelost			

Per fase van het project wordt aangegeven welke cellen op 0 moeten staan (waarmee het maturity-niveau aangegeven wordt)

Bijvoorbeeld bij het begin van een testfase moeten alle L1 problemen opgelost zijn, en alle L2 problemen in behandeling:

	L1	L2	L3
Nieuw	0	0	≥0
In behandeling	0	≥0	≥0
opgelost	≥0	≥0	≥0

Indien niet aan deze voorwaarden wordt voldaan, kan het aantal nog op te lossen problemen een indicatie zijn voor de volwassenheid van het product.

In praktijk wil het wel eens voorkomen dat diverse mensen binnen een project hun eigen lijstjes, bijvoorbeeld de maturity grid, bijhouden. Nadeel hiervan is dat niet alle lijstjes dezelfde resultaten geven. Als de één het lijstje rond het middaguur genereert, en de ander 's ochtends bij binnenkomst, kunnen de resultaten al verschillen. Ook is het maar de vraag of alle lijstjes op dezelfde zoekleutel gebaseerd zijn. De één vindt bijvoorbeeld dat je documentatie PR-en wel mee moet nemen, de ander niet.

Door de maturity grid centraal (door SCM) te laten genereren, en de uitkomst hiervan te distribueren onder de betrokkenen, wordt geborgd dat alle betrokkenen over dezelfde kengetallen praten.

Wat hebben we nu eigenlijk geleverd

Een andere vraag die de gemoederen vaak bezig houdt is "wat zit er nu eigenlijk precies in deze levering?"

Werd er in het verleden nog een rondje langs de ontwikkelaars gemaakt om te inventariseren wie nu wat geleverd had, tegenwoordig kan deze informatie door koppeling van een SCM systeem en een CR/PR tool relatief eenvoudig gegenereerd worden.

Waarom dan toch de onduidelijkheid over de inhoud van leveringen? Dit wordt veroorzaakt door een verschil in perceptie.

Een SCMer kijkt naar zijn SCM systeem, en herleidt de geleverde veranderingen naar de bijbehorende CRen/PRen. Op basis hiervan wordt bepaald wat er in de levering zit. Dit is ook wat geïdentificeerd wordt middels een zogeheten baseline, en wat vanuit SCM gezien wordt als "de waarheid".

Anderen, bijvoorbeeld een teamleider of integratiemanager, kijkt naar de status van de CRen/PRen in het CR/PR tool en baseert hier zijn conclusies op.

Idealiter zouden deze 2 lijsten overeen moeten komen. Echter, in praktijk komen er vaak verschillen voor.

Een goed voorbeeld hiervan is de ontwikkelaar die net voor hij op vakantie gaat nog even snel een probleem oplost in zijn eigen omgeving. De oplossing wordt als patch geleverd naar de testers, of in het ergste geval rechtstreeks naar de klant (buiten SCM om). De ontwikkelaar rondt netjes de administratie in de PR tooling af, dus vanuit de teamleider gezien lijkt alles ok.

In zijn haast om op vakantie te gaan vergeet de ontwikkelaar de oplossing te leveren naar SCM, met als gevolg dat bij de eerstvolgende officiële levering de klant zijn probleem weer terugkrijgt.....

Een ander voorbeeld is dat een ontwikkelaar een aantal PRen combineert in één levering, waardoor er vanuit SCM gezien slechts één PR geleverd lijkt te zijn, terwijl er volgens de administratie meerdere PRen opgelost zijn.

Door nu regelmatig (middels CSA) de administratie van het CR/PR tool te vergelijken met wat geleverd is volgens het SCM systeem kunnen problemen in een later stadium voorkomen worden. Op het moment van de BA zou de administratie sluitend moeten zijn.

Consistentie

Consistentie is één van de sleutelwoorden in de begrippen rondom SCM. Maar wat wordt nu eigenlijk bedoeld met consistentie, en hoe valt dit te meten?

Letterlijk betekent consistentie: "het vrij zijn van innerlijke tegenspraak"².

Er vanuit gaande dat een software project minimaal uit twee typen CI's bestaat, te weten documentatie en code, zijn er een drietal basisvormen van consistentie te onderscheiden:

Documentatie-documentatie:

Met een consistente set van documenten wordt bedoeld dat het gehele documentenbouwwerk een consistent geheel vormt. Met andere woorden: alle onderlinge referenties, inclusief versies, kloppen. Men wil immers zeker weten dat bijvoorbeeld de design specificaties gebaseerd zijn op de juiste versie van de systeem specificaties.

Een degelijk document beheerssysteem kan helpen om het documentenbouwwerk consistent te krijgen/houden, maar in praktijk blijkt het controleren hiervan veelal handwerk.

Eventueel is het controleren te automatiseren, maar dat staat of valt met het gebruik van goede templates en conventies voor de documentatie.

Documentatie-Code:

In een ideaal project zorgen stabiele requirements, samen met een goed requirement management proces en code inspecties ervoor dat er uiteindelijk opgeleverd wordt wat oorspronkelijk gevraagd was.

Maar wat nu als één van deze processen hiaten bevat?

Een deel kan opgelost worden door het (her-)inrichten van de SCM procedures en tooling zodat de gaten in de processen gedicht worden. Hierin schuilt echter het gevaar dat SCM eigenaar wordt van

² Bron: van Dale taalweb
(<http://www.vandale.nl/opzoeken/woordenboek/>)

problemen die eigenlijk helemaal geen SCM probleem zijn.

Als er bijvoorbeeld geen of een gebrekkig requirement management tool is, dan is dat voor een deel te ondervangen via procedures binnen het werkgebied van SCM. Hiermee wordt weliswaar een verbetering geïntroduceerd in de problematiek rondom requirement management, maar de basis van het probleem (geen of gebrekkige tooling) wordt niet aangepakt.

De taak van SCM zou zich hier moeten beperken tot een controlerende functie. Zeer gedetailleerde controles zijn zeer tijdsintensief, en kunnen het beste steekproefsgewijs gedaan worden. Vaak geeft dit al een indicatie of de relatie tussen de documentatie en code in orde is.

Er zijn ook enkele simpele controles te bedenken, die eenvoudig te automatiseren zijn. Stel dat er een wijziging plaatsvindt op een requirements-document, dan ligt het in de lijn der verwachting dat hierna een wijziging volgt op de ontwerpspecificaties en de code. Gebeurt dit niet, dan kan dit een signaal zijn dat er een hiaat in het proces zit.

Code-Code

Bij ontwikkelprojecten is het belangrijk om ook aandacht te besteden aan de consistentie van de geleverde code.

Hierbij dient vooral gekeken te worden of het geheel (zowel ontwikkelen, bouwen als testen) gebaseerd is op dezelfde omgeving. Denk hierbij vooral aan versies van compilers, libraries en standaarden, maar ook van interfaces tussen de diverse subsystemen in een groot en complex softwareproduct. Deze vorm van controle is vooral belangrijk als het eindproduct bestaat uit diverse software componenten die door verschillende partijen aangeleverd wordt. Maar ook binnen een eigen locale omgeving is het van belang dat alle partijen gebruik maken van dezelfde versies van software en interfaces.

Een praktijkvoorbeeld wat er mis kan gaan als de ontwikkel- en bouwomgeving niet consistent zijn:

De ontwikkeling van de software vindt plaats in een JAVA omgeving. Standaard gebruikt iedereen versie X. Echter, één van de ontwikkelaars heeft een leuke website gevonden, maar om die goed te kunnen bekijken heeft hij een nieuwere JAVA versie nodig. Dus de ontwikkelaar

download de nieuwste versie en installeert deze op zijn PC. Vervolgens gaat hij verder met het ontwikkelen van zijn code, maar nu (onbedoeld) gebaseerd op de nieuwste JAVA versie. En passant voegt hij nog even wat stukjes code toe gebruik makend van de laatste nieuwe features van JAVA.

Alle lokale testen gaan goed, dus de software wordt geleverd naar SCM om opgenomen te worden in het product. Echter, wat blijkt: SCM krijgt de nieuwe code niet gecompileerd. De oorzaak bleek uiteindelijk in het verschil van JAVA versie te zitten.

De toegevoegde waarde

De toegevoegde waarde van CSA en BA is tweeledig.

Aan de ene kant heeft project management hiermee een meetinstrument in handen waarmee betere controle op de ontwikkeling van het product mogelijk is.

Aan de andere kant kunnen afwijkingen ten opzichte van de verwachtingen aanleiding zijn tot verbeteringen in het ontwikkelproces.

De eerste baseline-audit bij het eerstgenoemde voorbeeld hierboven (analyseren bouwresultaten) liet zien dat er een groot aantal oude interfaces in gebruik was aan het eind van het project, waar het vervolgproject aanzienlijke (lees onverwachte) hinder van ondervond.

Naar aanleiding van de baseline audit is er vervolgens een procedure gekomen hoe om te gaan met het verwijderen van interfaces tijdens het project, waardoor vervolgprojecten nu geen last meer hebben van oude interfaces uit het verleden.

Tot slot

Kijkend naar CMMi, vallen CSA en BA onder Configuration Management. Dit wil overigens niet per definitie zeggen dat ze ook uitgevoerd dienen te worden door een (Software) Configuration Manager.

Iemand die veel doet aan metrieken, en weet hoe de benodigde gegevens uit de SCM systemen gehaald moeten worden kan bijvoorbeeld ook een zeer geschikte persoon zijn om dit soort rapportages te maken.

Belangrijk is vooral dat men weet wat men aan het meten is en hoe deze gegevens geïnterpreteerd dienen te worden. Uit de tools die de Software Configuration Manager doorgaans tot zijn beschikking heeft, zijn talloze gegevens te halen die gebruikt kunnen worden om een invulling aan CSA en BA te geven, vandaar dat deze taken vaak bij SCM liggen.

Om, naar aanleiding van CSA en BA, ook te kunnen verbeteren, is het zinvol iemand met

kennis van processen en verbeteringen te betrekken tijdens het maken van de rapportages.

Hieruit blijkt dan ook dat de disciplines Metrieken, Quality Assurance, Software Process Improvement en SCM soms meer met elkaar gemeen hebben dan men misschien zou vermoeden.

Pascal van Kempen
Software Configuration Manager bij Topic Embedded
Systems BV
Pascal.van.Kempen@topic.nl

■ Infotenties



“Regie bij Outsourcing met BiSL”

Datum/tijd: 6 maart 2008
15.30 uur - 20.00 uur
Locatie Getronics PinkRoccade,
Marconibaan 5, 3439 MR
Nieuwegein

Wat zijn uw overwegingen om te komen tot outsourcing? Hoe vult u dit in en hoe houdt u de regie hierover?

Dat zijn de vragen die aan orde komen in onze eerste themabijeenkomst dit jaar, op 6 maart van 16.00 - 20.00 uur in Nieuwegein.

Jan Hendriks van Quint geeft een schets van marktontwikkelingen in Regie & Outsourcing, waarna Rein van Vugt van Fortis zijn overwegingen met ons deelt om wel - of juist niet - over te gaan tot outsourcing. Ook bij de Immigratie en Naturalisatie Dienst (IND) is het vraagstuk van regie bij outsourcing actueel. Arnold de Leeuw van IND vertelt over zijn praktijkervaringen.

Debatleider Frederik Schukken van Quint zal de zaal tenslotte meenemen in een debat over de 'waarde van BiSL als framework bij het inrichten van een regieorganisatie'.

De themabijeenkomsten zijn gratis toegankelijk. U kunt zich aanmelden bij andre.smulders@ordina.nl.

www.aslbislfoundation.org



Eén-daagse training

CMMI for Acquisition Supplement for Introduction to CMMI Version 1.2

Datum/tijd: 25 februari 2008
Locatie SEI Germany, Frankfurt

This one-day course introduces acquisition managers and practitioners, appraisal team members, and process group (e.g., SEPG, EPG) members to CMMI® fundamental concepts related to acquisition. CMMI for Acquisition (CMMI-ACQ) is a CMMI model designed for use in managing a supply chain by those who acquire, procure, or otherwise select and purchase products and services for business purposes. This CMMI model focuses on acquirer processes and contains best practices that address activities for initiating and managing the acquisition of products and services that meet the needs of the customer. Some types of acquisitions that would benefit from using CMMI-ACQ include government acquisition, supply chain management, procurement, and outsourcing.

Call for papers
11^e SPIder conferentie
Thema: TBD
www.spiderconferentie.nl
Insturen voor 21 maart a.s.



SOFTWARE & SYSTEMS QUALITY CONFERENCES International | 2008

Datum/tijd: 15 – 18 April 2008
Locatie Düsseldorf, Duitsland

the programme for this year's Software & Systems Quality Conference, taking place 15th – 18th April 2008 in Dusseldorf (Germany), is online now.

As in recent years the conference will offer the delegates a combination of tutorials, presentations and workshops. As well as this there will also be keynotes held by top-class presenters from politics and economy.

We are glad to welcome the following Keynote-Speakers:

Dr. Tarek Kamel - Minister of Communications and Information Technology / Arab Republic of Egypt

Frank Ridder - Research Director Gartner Deutschland

Hans-Joachim Hentschel – Member of the Board Zürich Beteiligungs-Aktiengesellschaft and

Marc Gallagher - Eddie Jordan's Formula One Team

Should you be able to book now an Early Bird Discount will be granted. All bookings received no later than 23rd February 2008 will get a 15% discount.

Further information about Software & Systems Quality Conference can be found at www.sqs-conferences.com/de/index.htm.

Should you have questions or need further information please contact us at +49 (0)2203/9154-81 or team@sqs-conferences.com.

We are looking forward to seeing you in Dusseldorf!



■ Waarom je geen architect moet worden

drs. Ron L.J. van den Burg en Sjaak Laan

Veel beheerders, ontwikkelaars en engineers zien een architectuurfunctie als een stap voorwaarts in hun carrière. Een stapje hoger op de ladder naar succes. Maar zoals er geboren leiders zijn en mensen met een geboren creatieve geest, zijn ook architecten, engineers, projectleiders en managers geboren voor hun vakgebied. In dit artikel wordt het werkveld van ict-ers in vier activiteitstypen ingedeeld. Deze activiteitstypen onderscheiden zich door verschillen in profiel van de gedragskenmerken die nodig zijn voor de succesvolle uitvoering van die activiteiten. De schrijvers betogen dat een indeling naar activiteitstypen ict-ers inzicht verschaft een juiste carrièrepad te kiezen.

Marco is een ontwikkelaar die door hard werken en het volgen van diverse trainingen steeds meer in de belangstelling van zijn manager is komen te staan. Hij is zo langzamerhand het boegbeeld van de ontwikkelafdeling. Bij nieuwe projecten wordt hem stevast om zijn mening gevraagd. Het is Marco's droom op te klimmen tot Solutions Architect. Hij komt echter steeds meer tot de ontdekking dat de aansluiting met andere mensen in projectteams lastiger wordt. Voorheen kon hij altijd vertrouwen op zijn grondige kennis van zijn vakgebied, en had hij het idee dat hij hierin niet te overtreffen was. Nu hij meer en meer aan het begin van projecten wordt betrokken en geacht wordt de grote lijnen uit te zetten, gaat hij de diepgaande kennis met haar finesses missen. Marco voelt zich ongelukkig worden in zijn nieuwe rol. Het lijkt wel of hij de aansluiting niet kan vinden.

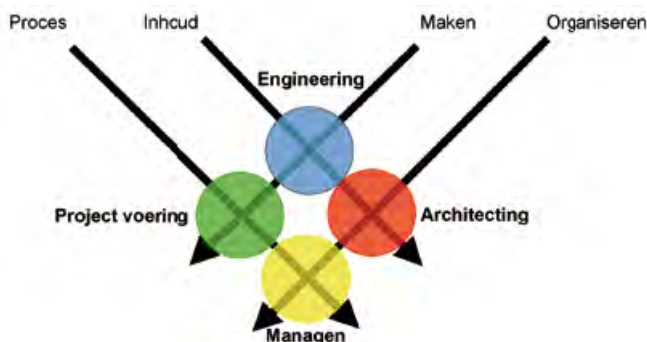
John is projectleider. Eerst van kleine projecten, maar in de loop van de tijd krijgt hij de verantwoordelijkheid over steeds grotere projecten. Het gaat John makkelijk af. Als er een vacature ontstaat als ict--manager, solliciteert John. Het is immers een grote stap voorwaarts in zijn carrière. John wordt aangenomen, maar helaas, na twee jaar is John afgebrand. Het leiden van projecten was weliswaar stressvol, maar voor hem goed vol te houden. Het managen van een afdeling blijkt toch een andere discipline. Hij voelt zich ongelukkig in zijn nieuwe rol. Waarom voelen Marco en John zich zo ongelukkig?

Het hoe, wat en waarom van architecten
In de literatuur wordt veel gesproken en gediscussieerd over wat een architect nu eigenlijk is. Wat hij zou moeten doen en waar hij verantwoordelijk voor is. Rijsenbrij [1] geeft een scheiding aan tussen architecten en engineers. Hij geeft aan dat er vier soorten

architecten zijn, te weten: enterprise architecten, domeinarchitecten, applicatie-architecten en werkuurarchitecten. In zijn eerdere publicatie [2] was er ook nog sprake van technische infrastructuurarchitecten, maar hiervan heeft hij later gesteld dat deze mensen vooral engineers zijn. Volgens Rijsenbrij volgen engineers en architecten verschillende werkwijzen. Architecten visualiseren wat de klant graag zou willen, terwijl engineers iets op een zo efficiënt mogelijke, goedkope en eenvoudige manier willen bouwen. Architecten maken het globale ontwerp, en engineers het detailontwerp. Hoewel de taak van engineers hierbij niet onderschat wordt, wordt toch impliciet de boodschap uitgedragen dat architecten zich op een ander (hiërarchisch) niveau bevinden dan engineers.

Ook Microsoft [3] definieert meerdere architectrollen. En ook hier wordt het verschil tussen architecten en engineers vooral gezocht in wat een architect geacht wordt te doen. Hiermee wordt de indruk gewekt dat men van een engineerrol kan doorstromen naar een architectenrol, wanneer een manier van werken wordt aangeleerd (middels trainingen en ervaring bijvoorbeeld).

In een persoonlijk gesprek met Jack Greenfield [4], Software Architect, Enterprise Tools bij Microsoft en eerste auteur van het boek Software Factories [5], blijkt dat de onoverkomelijke discussies over een indeling in soorten architecten op basis van de wat-vraag heel herkenbaar zijn. Zet een aantal architecten bij elkaar en ze worden het niet eens over wat een architect onderscheidt van een engineer.



Figuur 1: Vier type activiteiten

Onze stelling

Het belangrijkste kenmerk van een architect, engineer, projectleider en manager is wie hij is. Kennis kun je leren, ervaring bouw je op, maar gedrag is slechts in beperkte mate bij te stellen. Het zit bij wijze van spreken al in de genen van mensen in welke functie ze zich het gelukkigst zullen voelen. We kennen allemaal wel iemand van onze basisschool van wie als

kind al duidelijk was dat hij later een leidinggevende functie zou krijgen. Ook waren er duidelijk mensen te herkennen met oog voor detail. Als men deze kinderen later op bijvoorbeeld een reünie weer tegenkomt, blijken ze vaak een beroep te hebben gekozen dat aansluit bij de persoonlijkheidskenmerken die ze als kind al hadden. Mensen die geboren architecten zijn, pakken zaken van nature anders aan dan mensen die bijvoorbeeld geboren engineers zijn. Dit geldt ook voor het verschil tussen managers en projectleiders. Bij allemaal is de natuurlijke aanpak van groot belang voor hun succes. In dit artikel onderkennen we vier type activiteiten (zie figuur 1) op basis van het profiel van gedragskenmerken. Om de essentie van dit artikel goed over te brengen is het belangrijk even stil te staan bij de verschillen tussen personen, rollen, functies en de respectievelijke kenmerken. Waar we in de inleidende delen van dit artikel 'engineer' schreven (de rol) maar eigenlijk 'engineering' bedoelden, zullen we in het vervolg zorgvuldig met de verschillen omgaan. We beginnen met de activiteitstypen, waarna we ingaan op rollen en functies. Mensen zijn in hun werkwijze te onderscheiden in diegenen die organiseren en diegenen die maken. Daarnaast maken we een onderscheid tussen mensen die zich bezig houden met de inhoud of met het proces. Als we deze twee indelingen grafisch uitzetten in figuur 1, dan ontstaan op de 4 kruispunten 4 typen activiteiten. Voor deze 4 activiteiten hanteren we de labels engineering, projectvoering, architecting en managen. Er zijn natuurlijk managers met een grote inhoudelijke kennis. Dit is niet in tegenstelling met onze beleving. Het is waarschijnlijk dat we hier te maken hebben met personen met een natuurlijke activiteitstype architecting in een managementfunctie. De verschillen tussen de makende – en de organiserende activiteitstypen zijn nader uitgewerkt in tabel 1. Gedragsprofiel In de meeste modellen wordt een typering gehanteerd naar kennis en ervaring, het wat. De bovenstaande activiteitstypen geven echter een typering op basis van gedragskenmerken, het hoe. Veel mensen zien een managementfunctie als een carrièrestap na een projectleiderfunctie, of een functie als architect als dé groei vanuit een engineeringfunctie. Voor mensen met als dominant activiteitstype engineering of projectvoering kan dit slecht uitpakken.

Groei en carrièreperspectief zit wat ons betreft vooral in de mate van verantwoordelijkheid die men in een bepaalde rol heeft. Het is dan wel zaak om de beoordelingscriteria af te stemmen op de activiteitstypen. Belangrijk is te constateren dat ook mensen met engineeringactiviteiten opereren op verschillende niveaus en tot op het hoogste niveau kunnen doorgroeien zonder zichzelf

geweld aan te doen in onnatuurlijke rollen. In gangbare modellen kunnen meerdere rollen door één persoon worden ingevuld. Wij stellen de vraag: “wat is het belangrijkste activiteitstype dat nodig is voor de rollen?” Als de rollen een verschillend activiteitstype vereisen, dan voorzien we grote problemen bij het vervullen door één en dezelfde persoon. Bestaanden modellen die kijken naar persoonlijkheidstypen zijn de Big Five [7,8] en de Myers-Briggs Type Indicator [9]. In de Big Five en de MBTI herkennen we echter geen sterke correlatie met de IT-rollen en IT-functies. De Big Five helpt ons dus niet in het onderscheid maken tussen bijvoorbeeld engineering en architecting. We herkennen wel het Forer effect [10] en mede daarom hebben we gezocht naar een typering [a] waarvan het ene eind van de schaal ‘net zo mooi’ is als het andere eind. Bij het maken van de bovenstaande tabel is zorgvuldig geprobeerd om waardeoordelen te voorkomen [b]. De kenmerken aan de linkerkant zijn net zo belangrijk en waardevol als die aan de rechterkant. We beseffen dat we ons op het werkgebied van de arbeids- en organisatiepsychologie (meer in het bijzonder personeelspsychologie) begeven.

rekening te houden met verschillende persoonlijkheidstypen in een team, bijvoorbeeld de Belbin [11] teamrollen. We zien de toepassing van deze Belbin rollen nauwelijks in de praktijk en ervaren dat de vier activiteitstypen veel beter aansluiten bij de praktijk. Misschien wel het belangrijkste voordeel van een indeling op basis van activiteitstypen is dat mensen zich als vanzelf thuis voelen in één van de vier typen en er geen autoriteit nodig is om te vertellen wie het predicaat ‘manager’ of ‘architect’ verdient. Wat ons betreft mag je jezelf architect noemen als je de kenmerken behorend bij het activiteitstype architecting bezit. We hebben het dan niet over de senioriteit, het activiteitstype kan zowel passen bij een aspiring architect (geen ervaring) als bij een senior enterprise architect. Iemand met een engineeringprofiel die begrijpt dat een rol als manager, projectleider of architect niet alleen andere kennis en ervaring vereist maar ook bijbehorende activiteitstypen, zal ook begrijpen dat een verandering van rol voor hem geen promotie zal betekenen, maar een onnatuurlijke en ongelukkige carrièrestap. We willen de lezer bewust maken dat het slecht functioneren van een medewerker niet alleen hoeft te worden veroorzaakt door het ontbreken van kennis en kunde, maar ook dat de functie niet bij zijn persoonlijke aanpak kan passen.

Tabel 1

Maken (Engineering / Projectvoering)	Organiseren (Architecting / Managen)
Werken met zekerheden vooraf.	Zekerheden gaandeweg tegen komen, graag willen werken in een onzekere omgeving; vertalen van richting en doelen naar structuur.
Werken vanuit beproefde structuren.	Herkennen van structuren en gaandeweg structuren aan brengen.
Efficiënt werken.	Effectief werken.
Oplossingsgericht denken.	Conceptueel denken.
Maken, produceren.	Organiseren & structureren.
Levert geld op.	Kost geld, maar bij afwezigheid kost het meer geld.
Zorgen dat de dingen goed (conform opdracht) gedaan worden.	Zorgen dat de goede dingen worden gedaan.
Tijd verdelen over alle problemen.	80% van de tijd besteden aan de top 10 van de problemen.
Werken met gefaseerde aanpakken.	Viewpoint hopping[6] toepassen.
Leggen hun expertise consistent en consequent uit.	Vershillende viewpoints gebruiken om het verhaal naar verschillende stakeholders te brengen
Helderheid en duidelijkheid zijn belangrijk zodat de resultaten van het werk bijdragen aan de te halen doelen.	Ruimte en vrijheid zijn belangrijk, dat is het speelveld wat wordt ingevuld.
Planmatig tijd indelen.	Gebruiken van inspiratiemomenten.
Zonder engineering werkt het niet of niet goed.	Zonder architecting heb je er niets aan.
Zonder projectvoering komt het er niet of veel te laat.	Zonder managing krijg je iets waar niemand om gevraagd heeft.

Voor diegenen die op deze gebieden kundig zijn moet dit artikel gezien worden als een uitnodiging vanuit de praktijk om aansluiting te zoeken. Wat onze typering duidelijk maakt is dat geen van de vier activiteitstypen kunnen ontbreken in een organisatie of project. Dit wordt geïllustreerd door de laatste kenmerken uit bovenstaande tabel: het werkt dan niet, het komt dan niet op tijd, je hebt er niets aan of niemand heeft er om gevraagd. Ook in de gangbare modellen wordt geadviseerd

Rollen

De activiteitstypen geven aan dat situaties op verschillende manieren aangepakt worden. De typen zijn niet alleen te zien als een eigenschap van personen, maar ook als de gedragskenmerken van rollen. Een titel van een rol dekt niet altijd de (activiteiten)lading. Neem bijvoorbeeld de opdracht om een project te leiden waarvan de requirements en het doel niet helder of zelfs onbekend zijn. Mensen die makende activiteitstypen als hun dominante

natuur hebben, zullen verzuchten dat het een onmogelijk project is. Daarentegen zullen mensen die organiserende activiteitstypen als hun dominante natuur hebben als het ware gelukkig worden van zo'n opdracht omdat het veel ruimte geeft en de creativiteit aanspreekt. Met andere woorden, er wordt weliswaar gevraagd om een projectleidersrol, maar de context vereist een organiserend activiteitstype. Geconfronteerd met een probleem dat moet worden opgelost, zal elk persoon de eigen aanpak als de natuurlijke ervaren. Bij het inrichten van een organisatie zijn het juist kiezen van de functie-rol mapping (geen rollen met tegenstrijdige vereiste activiteitstypen samenbrengen in één functie) en een juist specificeren van de vereiste kwalificaties van functies van groot belang. Dit is de reden dat de vier gedragsprofielen niet één op één zijn te mappen op rollen. Zo is programmamanagement iets heel anders dan projectmanagement.

Voor programmamanagement is het activiteitstype managing relevanter dan het activiteitstype projectvoering. Stofberg en Trienekens [12] beschrijven architectenactiviteiten als facilitating, formalizing and controlling. Volgens Trienekens staat de controlling architect relatief dicht bij de engineer en de facilitating architect relatief dicht bij de manager. Alhoewel dit vreemd kan overkomen is dit goed te verklaren doordat het activiteitstype engineering belangrijk is bij controlling architects en het activiteitstype managing belangrijk is bij facilitating architects. Het zou dan ook te eenvoudig zijn om een IT-functiehuis in vier delen op te delen op basis van de vier activiteitstypen. Willen mensen voldoening hebben in hun werk dan moeten ze acteren in hun natuurlijke rol. Voor leidinggevers is het inzicht in activiteitstypen belangrijk om mensen die van nature een architecting activiteitstype hebben een groeipad richting architectuur te bieden, ook al blijkt dat niet uit hun werkverleden. In plaats dat we terugkoppelen wat mensen slecht doen, onderkennen we dat de persoon in een tegennatuurlijke rol werkt en juist sterke eigenschappen voor een andere rol heeft. Groei...

Mensen groeien het best in functies en rollen die een activiteitstype vereisen die aansluit bij hun natuurlijke activiteitstype.

...horizontaal op je plek komen. Weinigen starten hun carrière in de rol of functie projectleider, manager of architect, maar veelal in een engineeringrol. Voor mensen met een engineeringprofiel is dit een natuurlijke rol, maar voor de overige mensen is het een 'vreemde' rol. Voor die mensen met een architectingprofiel is een overstap van een engineeringrol naar een architectenrol niet zozeer een groeipad, maar moet eigenlijk beschouwd worden als een 'thuiskomen waar

je hoort.' ...of horizontaal om alle activiteitstypen te ervaren. De manier om te groeien naar een 'completere' ict'er is om tijdelijk rollen te vervullen die persoonsvreemde activiteitstypen vereisen. Zo kan het voor het succesvol uitoefenen van een architectenfunctie nuttig zijn om een tijdje een projectleidersrol op zich te nemen om zijn projectmatige kwaliteiten op peil te brengen. Voor mensen aan het begin van hun carrière die nog op zoek zijn naar hun carrièrepad kan een onorthodoxe aanpak als projectleider een signaal zijn voor onderkenning van een architect in spe. In principe geldt voor alle functies dat een tijdelijk uitoefenen van rollen met andere activiteitstypen enorm helpt bij het begrijpen van mensen die zo'n ander activiteitstype als natuur hebben. De ervaring in de andere rol moet dan ook worden gezien als een verrijking van de persoon die de kwaliteit van de communicatie met de 'andere kant' verbetert. Als het aangeboren activiteitstype niet onderkend wordt ontstaat er een onjuiste veronderstelling dat projectleiders, managers of architecten 'beter' zouden zijn. De functie van engineer wordt dan 'slechts' als startfunctie gezien. Alle vier de activiteitstypen zijn echter nodig in een project of bedrijf.

Dat ict-carrières starten in de rol engineer verklaart het feit dat startsalarissen voor engineering rollen lager zijn dan die voor niet-engineering rollen. Dat betekent echter niet dat de eindsalarissen ook zouden moeten verschillen. Wij vinden dat in alle rollen voldoende groeiperspectief mogelijk moet zijn. Het zou in ieder geval onverstandig zijn op basis van (een perceptie van) een succesvoller carrièrepad een tegennatuurlijke functie te ambiëren. We zijn van mening dat het rekening houden met de natuurlijke aanleg van mensen en een bewuste situationele inzet van mensen in rollen, er voor zorgt dat mensen beter op hun persoonlijke kwaliteiten worden beoordeeld die ze wél hebben dan op de verwachte rol kwaliteiten die ze níét hebben.

Vervolg

Wij stellen dat het activiteitstype van een persoon onveranderlijk is, hoewel dit niet wetenschappelijk gestaafd is. Feedback hierop en liefst verwijzingen naar bestaand onderzoek wordt door ons hoog op prijs gesteld. Dit artikel mag ook worden gezien als een uitnodiging aan de professionele wereld van gedragswetenschappen en certificeringinstanties om te reageren. Uitbreidingen van het model moedigen wij aan. Mogelijke uitbreidingen van het model kunnen zijn:

- Het model toepassen op een grotere groep mensen voor het vergroten van de toepasbaarheid.
- Aanvullende gedragskenmerken voor verbetering van het model.

- Een 'Viva-test' waarmee met behulp van een vragenlijst bepaald kan worden welke de belangrijkste persoonskenmerken zijn van iemand, en in welke rol hij zich dus waarschijnlijk het meest thuis zal voelen.
- Een mapping van de vier gedragsprofielen op andere bestaande modellen om te kijken of er al iets vergelijkbaars is beschreven.
- Onderzoek om bijvoorbeeld te kijken hoe de verdeling is in aantallen per gedragsprofiel.

Ten slotte

Hoewel de in dit artikel gepresenteerde activiteitstypen zich concentreren op de vraag wie een architect, engineer, projectleider of manager eigenlijk is, blijkt dat veel engineers voor zichzelf een groeipad richting architectuur zien. Schoolverlaters krijgen over het algemeen geen architectenfunctie, omdat de ervaring hiervoor ontbreekt. Of mensen uiteindelijk architect moeten worden hangt vooral af van de manier waarop ze zaken aanpakken en de gedragskenmerken zoals in dit artikel zijn beschreven. Geboren architecten zullen daarom vanzelf komen bovendien. Als men de benodigde architectkenmerken niet heeft, dan zou het veel beter zijn te excelleren als engineer, en zich eventueel te specialiseren op een bepaald aspect hiervan. Engineers die architecten worden, maar er niet voor zijn geboren, zullen naar onze mening ongelukkig worden in de uitoefening van hun professie.

Voetnoten

a Het Forer-effect, ook Barnum-effect genoemd, is afkomstig uit de psychologie en betreft de neiging van mensen om vage en algemeen geldende uitspraken over de eigen persoon te accepteren als een rake, typerende omschrijving, zonder zich te realiseren dat diezelfde omschrijving voor bijna iedereen opgaat. Voorzitter, vormgever, plant, waarschuwer, bedrijfsman, brononderzoeker, groepswerker, afmaker.

b Met nadruk willen we aangeven dat we hier niet de indruk willen wekken "wie als een dubbeltje geboren wordt, wordt nooit een kwartje". Dat zou immers veronderstellen dat engineers dubbeltjes zijn hetgeen we bestrijden.

Referenties

- [1] Kanttekeningen bij Architectuur in de Digitale Wereld (versie nulpuntzes); Prof. Dr. Daan Rijsenbrij; 2005
- [2] Architectuur in de Digitale Wereld (versie nulpunt drie); Prof. Dr. Daan Rijsenbrij; 2004
- [3] What is an Architect; Microsoft; <http://msdn2.microsoft.com/en-us/skyscraper/bb401007.aspx>; 2007
- [4] Jack Greenfield's Blog can be found at <http://blogs.msdn.com/jackgr/>
- [5] Software Factories; Jack Greenfield, Keith Short, Steve Cook, Stuart Kent; 2004
- [6] Architectural Reasoning, course book; Gerrit Muller; 2005

[7] Factors in the natural language of personality: re-analysis and comparison of six major studies; Digman and Takemoto-Chock; 1981

[8] Personality in adulthood; McCrae and Costa; 1990

[9] Introduction to Type: A Description of the Theory and Applications of the Myers-Briggs Type Indicator; Myers; 1990

[10] The fallacy of personal validation: A classroom demonstration of gullibility; Forer; 1948

[11] Management Teams – Why They Succeed or Fail; Belbin; 1981

[12] Architectuur, een complot tegen het succes van de onderneming?; Stofberg en Trienekens; TIEM 13; mei 2006

■ System Architecture: The Silver Bullet?

Gerrit Muller (*Embedded Systems Institute*)

Distribution

This article or presentation is written as part of the Gaudí project. The Gaudí project philosophy is to improve by obtaining frequent feedback. Frequent feedback is pursued by an open creation process. This document is published as intermediate or nearly mature version to get feedback. Further distribution is allowed as long as the document remains complete and unchanged.

All Gaudí documents are available at:

<http://www.gaudisite.nl/>

version: 1.8 status: preliminary draft 10th April 2007

Abstract

Many people jump onto systems architecting methods with the false expectation that these methods will solve most of their problems. A major reason to do this is uncertainty. The real world, unfortunately, is full of uncertainties, and systems architecting can help in dealing with these many uncertainties. However, systems architecting will not make uncertainties disappear, neither will it prevent effort to be spent on unexpected issues.

Silver Bullets do not exist

The critical success factors for applying system architecture methods are described.

1 Introduction

The expectation level with respect to processes in general and the system architecture process in particular can vary from skeptical to blind faith. The skeptics have experienced that horrible specifications and designs can be pursued under the grand name of Architecture. The followers with blind faith are at the opposite end of the spectrum, their believe in processes inhibits them from seeing the limitations and constraints from the processes applied.

The central message of this Intermezzo is:
Silver Bullets do not exist.

This Intermezzo intends to set realistic expectation levels with respect to the System Architecture Process, and describes the ingredients for successful application.

2 Why System Architecture?

System Architecting is a means to create systems efficient and effective, by supplying overview, by guarding consistency and integrity, and by balancing. In other words the System Architect helps the development team to find its way in a rather complex, dynamic and uncertain world.

From psychological point of view people apply their own survival mechanisms, when they perceive a threat. One of the most common survival mechanisms is The Quest for Certainty, see subsection 2.1.

Unfortunately System Architecting will never remove all uncertainties, see subsection 2.2. The application of a system architecture process can help in the risk management, amongst others by prevention, and by minimizing impact.

Successful application of system architecture is far from trivial, section 3 describes **how** the System Architecture Process should be applied to meet the goals of efficiency and effectively.

2.1 The Quest for Certainty

This section provides a caricatural view on human behavior based on a free interpretation of the Maslow Hierarchy of Needs, as discussed for instance in [1]. This exaggerated view matches with the security needs in the lower Maslow Hierarchy. Note that less defensive behavior can be triggered by needs in the higher layers, were words such as adventurous en explorative are being used.

The majority of people, including managers and engineers, have a need for certainty. Their ideal is to have stable, unchangeable sets of specifications, schedules et cetera. This (hopefully) isolates them from the nasty surprises of reality see table 1.

incompetent people
human mistakes
lack of collaboration or synergy
misunderstanding or miscommunication
changing markets
fast moving competition
unforeseen physical, chemical, mechanical properties
mother nature (illnesses, floods)

Table 1: Nasty Surprises of Reality

Unfortunately these nasty surprises are a fact of life. Our human capability to control these phenomena is quite limited.

Risk management can help to be more robust. However risk management certainly does not remove these phenomena and it also does not reduce the consequences to zero. Risk management balances probability, effect, and cost.

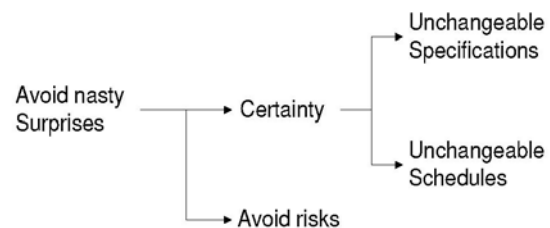


Figure 1: Personal key driver to avoid nasty surprises

People with a need for certainty are willing to accept any method or process which promises certainty. In other words *certainty* appears to be their personal key driver. It is better to rephrase this key driver as *to avoid* nasty surprises, which is closer to the internal motivation at the one hand and which gives a handle later on to manage the expectations. Figure 1 visualizes these drivers.

2.2 Disclaimer; Setting the Expectations to a realistic level

The Gaudí Project will not deliver a Plug-and-Play System Architecture Process. System architects which have read all the articles and followed the course will not automatically be successful.

The Gaudí project will deliver a large set of consistent background material for system architects. This material ranges from process and architecture principles, providing insight and understanding, to more specific how-to's which provide more directly applicable guidelines.

The competent system architect will use the material by customizing it to the specific problem to be addressed. At the same time the system architect will have to interact with the environment to share this customized way of working.

Whenever the material is applied literal, this is a strong indication that the organization and the system architect do not work explicit enough on the way of working.

3 How: Critical Success Factors

Ingredients for an effective application of a system architecture process are shown in table 2.

Know-How
Common Sense
Pragmatics
Critical attitude
Social skills
Drive
Vision

Table 2: Critical Success Factors for an effective application of a System Architecture Process

No method or process will function without these critical success factors. A process can not be used as substitute for know how or common sense.

3.1 Know-How

The core of the system architecture work is know-how, ranging from pure technology know-how to application and business know-how. Active control on a broad basis is a prerequisite for a system architect.

3.2 Common Sense

Most problems encountered during Product Creation require common sense to solve them. Mechanistic approaches severely limit the solution space, resulting in complex solutions. System architects are capable of "lateral" thinking, allowing solutions in previously unexpected directions.

3.3 Pragmatics

The holistic approach can easily derail in a sea of seemingly conflicting requirements and viewpoints. The system architect needs a significant amount of pragmatism to be selective and focused, while being holistic in the back of his mind.

3.4 Critical attitude

Clear diagrams, tables with facts and smooth presentations give the impression of high quality and increase the confidence. However these same diagrams, tables and presentations conceal the forgotten, misinterpreted, or underestimated facts. The system architect must always be alert, for instance by asking questions as shown in table 3.

Do we address the right problem or requirement?
Is the customer/user on-board?
Is this design adequate?
Consists the input data from facts, wishes or ideas?
Do we need so many people for the implementation?
Does this process or organization fit the problem?

Table 3: Critical Attitude: Examples of questions to be asked by the System Architect

3.5 Drive

A good system architect has a passion for his architecture, it has an emotional value.

An architect which is working entirely according to the book, obediently going through the motions, will produce a clinical architecture without drive or ownership.

Good architectures have an identity of themselves, which originate in the drive of the architect. Such an architecture is an evolving entity, which is appreciated by the stakeholders.

3.6 Vision

The system architect needs to have a vision to be able to provide direction. A vision enables an evolution of existing architectures to desired architectures. Having vision is not trivial, it requires a good understanding of needs (the problem) and means (the solution) plus the trends (opportunities and threats) in both needs and means.

4 Summary

The one sentence summary of this intermezzo is: Silver bullets do not exist. Table 4 gives a bullet-wise summary.

Most people want to avoid nasty surprises Most people are looking for certainty Silver Bullets do not exist
System Architecture is not a golden bullet Critical Success Factors: Know-How, Common Sense, Pragmatics, Critical attitude, Drive and Vision

Table 4: Summary

5 Acknowledgements

Hans Gieles suggested improvements to increase the cohesion and the red line in this Intermezzo.

Henk Obbink and Angelo Hulshout and many others pointed out that "The Golden Bullet" should have been "The Silver Bullet", which has been changed finally. Adriaan van den Brand pointed out the forgotten change of golden into silver.

Eugene Ivanov pointed out that evolution aspects were missing. The result is the addition of vision as critical success factor.

Steve R. Nanning indicated unclarity in the abstract and criticized the too negative phrasing of the *Quest for Certainty*. Stephen Boggess added a question to the list in Section "Critical Attitude". Ning Lu added more surprises and critical success factors.

References

- [1] W. Huitt. Maslow's hierarchy of needs. Educational Psychology Interactive. Valdosta, GA: Valdosta State University, 2004.
- [2] Gerrit Muller. The system architecture homepage. <http://www.gaudisite.nl/index.html>, 1999.

■ What's Wrong With Agile Software Methods?

Tom Gilb

Some Agile Principles, Values, and Cases to Encourage Quantification.

Abstract

- Agile Software Methods [AA] have virtually no **Quantified focus** on the quality and performance levels of the software. Particularly, no quantification of the main reasons why the project was funded (i.e. savings, more business, better service).
- This means you cannot directly control the main benefits for users and stakeholders.
- This means that you cannot really control the corresponding costs of getting the main benefits. You cannot estimate them and you won't get a budget for them.
- It means the current conventional agile methods are not suitable for industrial products; where you cannot simply deliver the 'functionality', but you must compete to deliver all sorts of qualities and performance attributes.
- I would like to suggest that we must consciously add, to agile concepts, the explicit notion of quantification of the business-critical variable benefits. These benefits can loosely be called qualities and performance characteristics. I will focus on quality characteristics, because that is where most people have problems of quantification.
- Let me be clear here that *quantification* (what do we want to control?) is only a first step in getting control. The next steps are *estimation* (how good do we think our solutions are?) and *measurement* (how good were the solutions in practice?), based on that quantification.

The Principle Of 'Quality Quantification'

- All qualities can be expressed quantitatively,
 - 'qualitative' does not mean unmeasurable. (Gilb)

- "In physical science the first essential step in the direction of learning any subject is to find principles of numerical reckoning and practicable methods for measuring some quality connected with it.*
- I often say that when you can measure what you are speaking about, and express it in numbers, you know something about it.*

- But when you cannot measure it, when you cannot express it in numbers, your knowledge is of a meagre and unsatisfactory kind.*
- It may be the beginning of knowledge, but you have scarcely in your thoughts advanced to the state of Science, whatever the matter may be."*

○ Lord Kelvin, 1893 From <http://zapatopi.net/kelvin/quotes.html>

Quality is characterized by these traits

- A quality describes 'how well' a function is done.
- Qualities each describe the *partial effectiveness* of a function (as do all other performance attributes).
- Relevant* qualities are either *valued* to some degree by *some* stakeholders of the system - or they are not relevant.
- More* quality is generally *valued* by stakeholders; especially if the increase is free, or lower cost, than the *stakeholder-perceived value* of the increase.
- Quality attributes can be *articulated* independently of the particular means (designs, architectures) used for reaching a specific quality level; even though all quality levels *depend* on the particular designs used to achieve quality.
- A particular quality can potentially be a described in terms of a *complex* concept, consisting of multiple elementary quality concepts. Love is a many-splendored thing!
- Quality is *variable* (along a definable scale of measure: as are all scalar attributes).
- Quality* levels are capable of being specified *quantitatively* (as are *all* scalar attributes).
- Quality levels can be *measured* in practice.
- Quality levels can be *traded off* to some degree; with other system attributes valued more by stakeholders.
- Quality can *never* be *perfect* (no fault and no cost) in the real world.
- There are *some valued levels* of a particular quality that may be *outside the state of the art*; at a defined future time and circumstance.
- When quality levels increase towards *perfection*, the resources needed to support those levels tend towards *infinity*.
- Source: CE, 'Quality' in Concept Glossary.

What can we do better in Agile Development (or 'at all'), if we **quantify** quality ideas?

- Evaluation of solutions / designs / architectures against the quantified quality requirements (Design-Impact Estimation, more later).
- Test and measure - the degree to which solutions meet - quality and cost

expectations (those set when solutions were chosen).

- Measure *evolutionary project progress towards quality goals*.
- And get early & continuous improved estimates for time to completion.
- Communicate quality goals much better to all parties.
- (users, customers, developers, testers, lawyers).
- Contract for results.
- Pay for results only (not effort expended).
- *Reward* teams for *results* achieved.
- *Motivate* technical people to *focus* on real business *results*.
- *Simplify* requirements.
- (the top few quantified- everything else is design)
- Collect numeric data about designs, processes, organizational structures.
- To learn and use in future. To assert you have improved, to get funding for improvement, to benchmark with similar organizations.
- Permits systematic corporate or academic research about a development environment.
- Get some *facts* as basis for process efficiency improvement!

Ten Agile Principles: to avoid bureaucracy and give creative freedom.

1. Control projects by quantified critical-few results. 1 Page total! (not stories, functions, features, use cases, objects, ..).
2. Make sure those results are business results, not technical.
 - Align your project with your financial sponsor's interests!
3. Give developers freedom, to find out *how* to deliver those results.
4. Estimate the impacts of *your* designs, on your quantified goals.
5. Select designs with the best impacts for their costs, do them first.
6. Decompose the workflow, into weekly (or 2% of budget) time boxes.
7. Change designs, based on quantified experience of implementation.
8. Change requirements, based in quantified experience, new inputs.
9. Involve the stakeholders, every week, in setting quantified goals.
10. Involve the stakeholders, every week, in *actually* using increments.

Ten Agile Values

- Simplicity
 1. Focus on real stakeholder values.
- Communication
 2. Communicate stakeholder values quantitatively.
 3. Estimate expected results and costs for weekly steps.

- Feedback
 4. Give deployment, of your understanding, weekly, to stakeholders, in their environment.
 5. Measure all critical aspects of the deployment.
 6. Analyze deviation from estimates.
- Courage
 7. Change plans to reflect weekly learning.
 8. Immediately implement valued stakeholder needs, next week.
 - *Don't wait, don't study (analysis paralysis), don't make excuses.*
 - *Just Do It!*
 9. Tell stakeholders exactly what you will deliver next week.
 10. Use any design, strategy, method, process that works quantitatively well - to get your results.
 - Be a systems engineer, not a just programmer (a 'Softcrafter').
 - Do not be limited by your craft background, in serving your paymasters.

Agile Project Management Policy

- The project manager, and the project, will be judged exclusively on.
 - The relationship of progress towards achieving the goals.
 - Versus the amounts of the budgets used.
 - The project team will do anything legal and ethical to deliver the goal levels within the budgets.
 - The team will be paid and rewarded for.
 - Benefits delivered.
 - In relation to cost.
 - The team will find their own work process and their own design.
 - As experience dictates, the team will be free to suggest to the project sponsors (stakeholders) adjustments to 'more realistic levels' of the goals and budgets.

The Simplest and Best Agile Project Method

- **Process Description**
 1. Gather from all the key stakeholders the top few (5 to 20) most critical goals that the project needs to deliver.
 - Give each goal a reference name (a tag).
 2. For each goal, define a scale of measure and a 'final' goal level.
 - For example: *Reliable: Scale: Mean Time Before Failure, Goal: 1 month.*
 3. Define approximately 4 budgets for your most limited resources.

- (For example, time, people, money, and equipment).
- Write up these plans for the goals and budgets.
 - (*Try to ensure this is kept to only one page*).
 - Negotiate with the key stakeholders to formally agree the goals and budgets.
 - Plan to deliver some benefit.
 - (That is, progress towards the goals).
 - In *weekly* (or shorter) increments (Evo steps).
 - Implement the project in Evo steps.
 - Report to project sponsors after each Evo step (weekly, or shorter) with your best available estimates or measures, for each performance goal and each resource budget.
 - On a *single* page, summarize the *progress to date* towards achieving the goals and the costs incurred.
 - When all Goals are reached: 'Claim success and move on'.
 - Free remaining resources for more profitable ventures.

Case Study of Conformat Product Illustrating these Improved Agile Ideas in Practice.

Evolutionary project planning

We collected the most promising solutions and included them in an Evo plan, which was expressed by using an Impact Estimation (IE) table (See Tables 2 and 3). The solutions were evaluated with respect to *value* for clients versus *cost of implementation*. We chose the ones with the highest value first. Note that value can sometimes be defined as *removing risks* by implementing technically challenging solutions early.

The IE table is our tool for controlling the qualities, and delivering improvements to real stakeholders, or as close as we can get to them (for example, delivering to our support people, who use the system daily and so can act as 'clients').

Recoding:

Type: Solution [Conformat 8.5].

Description: Make it possible to recode a variable, on the fly, from Reportal.

Estimated effort: 4 days.

Figure 2. A brief specification of the solution, 'Recoding'

Design Idea: Step 9 - Recoding				
Requirements	Estimated Scale Impact	Estimated % Impact	Actual Scale Impact	Actual % Impact
Objectives				
Usability Productivity 65 <-> 25 minutes Past: 65 minutes. Tolerable: 35 minutes. Goal: 25 minutes.	65 - 20 = 45 minutes	50%	65 - 38 = 27 minutes	95%
Resources				
Development Cost 0 <-> 110 days	4 days	3.64%	4 days	3.64%

Table 2: Here is a simplified version of the IE table for Evo Step 9, 'Recoding' of the MR project. Notice the definitions for the requirements and costs. The Planguage keyed icon '<->' means 'from baseline to target value. Step 9 alone moved the Productivity value to 27 minutes, or 95% of the way to the target level

The Evo method's impact on Conformat's product qualities

The Evo method's impact on Conformat's product qualities is *not* actually measured statistically, by doing a scientifically correct large-scale survey, although we are currently considering this. The impact described in this paper for Conformat 8.0 ('Past') is based on internal usability tests, productivity tests, performance tests carried out at Microsoft Windows ISV laboratory in Redmond USA, and from direct customer feedback. See Table 1 - only highlights of the impacts are listed here. No negative impacts are hidden.

	A	B	C	D	E	F	G	BX	BY	BZ	CA
1											
2											
3		Current	Improvements		Goals						
4		Status									
5		Units	Units	%	Past	Tolerable	Goal	Units	%	Units	%
6					Usability.Replacability (feature count)						
7		1.00	1.0	50.0		2	1	0			
8					Usability.Speed.NewFeaturesImpact (%)						
9		5.00	5.0	100.0		0	15	5			
10		10.00	10.0	200.0		0	15	5			
11		0.00	0.0	0.0		0	30	10			
12					Usability.Intuitiveness (%)						
13		0.00	0.0	0.0		0	60	30			
14					Usability.Productivity (minutes)						
15		20.00	45.0	112.5		65	35	25	20.00	50.00	36.00
20					Development resources						
21			101.0	91.8		0		110	4.00	3.64	4.00

Table 3: Details of the real table, which was simplified in Table 2

Description of requirement/work task	Past	Status
Usability.Productivity: Time for the system to generate a survey	7200 sec	15 sec
Usability.Productivity: Time to set up a typical specified Market Research-report (MR)	65 min	20 min
Usability.Productivity: Time to grant a set of End-users access to a Report set and distribute report login info.	80 min	5 min
Usability.Intuitiveness: The time in minutes it takes a medium experienced programmer to define a complete and correct data transfer definition with Confirmit Web Services without any user documentation or any other aid	15 min	5 min
Workload Capacity.Runtime.Concurrency: Maximum number of simultaneous respondents executing a survey with a click rate of 20 seconds and an response time < 500 milliseconds, given a defined [Survey-Complexity] and a defined [Server Configuration, Typical].	250 users	6000

Table 4: Improvements to product qualities in Confirmit 8.5

These leaps in product qualities would not have been achieved without Evo. We have received many pleasant emails regarding these quality improvements from our customers. To give an example:

"I just wanted to let you know how appreciative we are of the new "entire report" export functionality you recently incorporated into the Reportal. It produces a fantastic looking report, and the table of contents is a wonderful feature. It is also a HUGE time saver."

On the *second* Release, Confirmit 9.0, the Vice President (VP) of Marketing proudly named the Evo development method on the FIRM website³, and the quantified results they can deliver to customers. I cannot recall a technical paper about development methods with anything similar - a line executive bragging about a development method!

"FIRM, through evolutionary development, is able to substantially increase customer value by focusing on key product qualities important for clients and by continuously asking for their feedback throughout the development period. Confirmit is used by the leading market research agencies worldwide and Global 1000 companies, and together, we have defined the future of online surveying and reporting, represented with the Confirmit 9.0."

Figure 3. Comments by FIRM's VP of Marketing, Kjell Øksendal

The *above* quote by the VP of Marketing summarizes executive satisfaction with the new development method on their website.

³ I invite the reader to imagine a real marketing VP saying "We just started using [CMMI/RUP] 3 months ago and I want to report some fantastic quantitative product improvements".... ☺

The detail below is from the same website product release.

News release
2004-11-29: Press Release from FIRM
New version of Confirmit increases user productivity up to 80 percent
NOVEMBER 29th, 2004: FIRM, the world's leading provider of online survey & reporting software, today announced the release of a new version of Confirmit delivering substantial value to customers including increased user productivity of up to 80 percent.
FIRM is using Evolutionary (EVO) development to ensure the highest focus on customer value through early and continuous feedback from stakeholders. A key component of EVO is measuring the effect new and improved product qualities have on customer value. Increased customer value in Confirmit 9.0 includes:
* Up to 175 percent more intuitive user interface*
* Up to 80 percent increased user productivity in questionnaire design and testing*
* Up to 1500 percent increased performance in Reportal and Panel Management*
Figure 4. Confirmit 9.0 release announcement from the FIRM website, http://www.firmglobal.com. It gives detail about the method and the quantified product results

See Tables 5, 6, 7 and 8 for some more detailed results from Confirmit 9.0, by product component.

Feedback from developers and project managers within FIRM R&D

Evo has resulted in increased *motivation* and *enthusiasm* amongst developers, because it opens up for *empowered creativity*. This means that the developers can determine their own design ideas, and are not subject to being dictated the design ideas by marketing and/or customers, who often tend to be amateur technical designers.

Daily product builds were introduced, which we called Continuous Integration (CI). Evo combined with CI, is a vehicle for innovation and inspiration. The developers get their work out onto the test servers, and receive feedback. Every week.

Authoring		
Product Quality	Description	Customer Value
Intuitiveness	Probability that an inexperienced user can intuitively figure out how to set up a defined Simple Survey correctly.	Probability increased by 175%
Productivity	Time in minutes for a defined advanced user, with full knowledge of 9.0 functionality, to set up a defined advanced survey correctly.	Time reduced by 38%

Reportal		
Product Quality	Description	Customer Value
Performance	Number of responses a database can contain if the generation of a defined table should be run in 5 seconds.	Number of responses increased by 1400%

Survey Engine		
Product Quality	Description	Customer Value
Productivity	Time in minutes to test a defined survey and identify 4 inserted script errors, starting from when the questionnaire is finished to the time testing is complete and is ready for production. (Defined Survey: Complex survey, 60 questions, comprehensive JScripting.)	Time reduced by 83% and error tracking increased by 25%

Panel Management		
Product Quality	Description	Customer Value
Performance	Maximum number of panellists that the system can support without exceeding a defined time for the defined task, with all components of the panel system performing acceptably.	Number of panellists increased by 1500%
Scalability	Ability to accomplish a bulk-update of X panellists within a timeframe of Z seconds.	Number of panellists increased by 700%
Intuitiveness	Probability that a defined inexperienced user can intuitively figure out how to do a defined set of tasks correctly.	Probability increased by 130%

Tables 5, 6, 7 and 8: The tables show some detailed results from Confirmat 9.0, by product component

Even though they embraced the method, there are parts of Evo that they found difficult to understand and execute at first. These included:

- Defining good requirements can be hard;
- It was hard to find meters (that is, ways of measuring numeric qualities, testing quality levels), which were practical to use, and at the same time measured real product qualities;
- Sometimes it takes more than a week to deliver something of value to the client;
- Testing was sometimes postponed in order to start the next step. Some of these test deferments were then not in fact done in later testing.

Lessons learned with respect to the method

Some of the lessons we learned after the trial period include:

- We need to have increased focus on feedback from clients. We will select the ones that are willing to dedicate time to us. *Internal* stakeholders (like sales and help desk staff) can give valuable feedback, but some *customer* interaction is necessary;
- Demonstrate new functionality automatically, with screen recording software or early test plans. This makes it

easier for internal and external stakeholders to do early testing;

- Tighter integration between Evo and the test process is necessary;
- 'Be humble in your promises, but overwhelming in your delivery'.

By May 2005 FIRM had made use of a 'Green Week' once monthly to focus on the evolutionary improvement of about 12 internal stakeholder qualities (like testability and maintainability).

Releases 9.5 and 10.0 continued this pattern of successful improvement to the product for the customers (as of November 2005).

CONCLUSIONS (by Trond Johansen, FIRM A/S).

The Evo method's positive impact on Confirmat product qualities has convinced us that Evo is a better suited development process than our former Waterfall process, and we will continue to use Evo in the future.

What surprised us the most was the method's power of focusing on delivering value for clients versus the cost of implementation. Evo enables you to re-prioritize the next development-steps based on the weekly feedback. What seemed important at the start of the project may be replaced by other solutions based on gained knowledge from previous steps.

The Evo method has high focus on measurable product qualities, and defining these clearly and testably requires training and maturity. It is important to believe that everything can be measured and to seek guidance if it seems impossible.

One pre-requisite related to the method for using Evo is an *open architecture*.

Another pre-requisite is *management support* for changing the work process, and this is important in any software process improvement initiative.

The concept of daily builds, CI, was valuable with respect to delivering a new version of the software every week.

Overall, the whole organization has embraced Evo. The release of Confirmat 8.5 showed some of Evo's great potential, and we will work hard to utilize it to the full in the future. In June 2004, we had Tom and Kai Gilb present a 4-day course to the whole R&D department and other related resources. By the end of November 2004, with the second release (9.0), we confirmed that the method can, consistently and repetitively, produce the results we need

to lead our industry. We now expect the next versions of Conformat will prove that we have matured in our understanding and execution of Evo. (Source Trond Johansen, FIRM A/S).

References

AA: <http://www.agilealliance.com/>

CE: Gilb05: Gilb, Tom, Competitive Engineering, A Handbook For Systems Engineering, Requirements Engineering, and Software Engineering Using Planguage, ISBN 0750665076, 2005, Publisher: Elsevier Butterworth-Heinemann.

FIRM: From Waterfall to Evolutionary Development (Evo)

Or, How we rapidly created faster, more user-friendly, and more productive software products for a competitive multi-national market

Trond Johansen, Head of Project Management, FIRM A/S, Trond.Johansen@firmglobal.com
Published in INCOSE 05 Proceedings. INCOSE.org. and EuroSPI, Trondheim 2004.

Also published in Torgeir Dingsøyr (Ed.): Software Process Improvement, 11th European Conference, EuroSPI 2004, Trondheim, Norway, November 10-12, 2004, Proceedings. Lecture Notes in Computer Science 3281 Springer 2004, ISBN 3-540-23725-9. http://www.conformat.com/news/release_20041129_conformat_9.0_mr.asp

E-mail: Tom@Gilb.com

URL: <http://www.Gilb.com>

Version Nov 9 2005

Published on the occasion of the Seventeenth Annual International Symposium of the International Council On Systems Engineering (INCOSE) 24 - 28 June 2007

Copyright © 2007 by Tom Gilb. Published and used by INCOSE with permission.

■ Call for papers 11e SPIDER Conferentie

Op 29 september 2008 vindt de jaarlijkse SPIDER Conferentie plaats. In de afgelopen jaren is gebleken dat presentaties uit de praktijk van SPI en QA professionals hoog gewaardeerd worden door de deelnemers. Heb je ervaring met SPI of QA, passend binnen het thema van de conferentie? Stuur dan de Call for Papers vóór 21 maart a.s. volledig ingevuld in.

Verandering, we hebben er dagelijks mee te maken. Steeds meer organisaties werken met teams, gaan agile, en gebruiken principes van lean development. Ze moeten voldoen aan allerlei standards en regels. Er wordt samengewerkt met leveranciers, of delen van het werk worden ge-outsourced of juist ge-insourced. Geen dag zonder dat er weer iets nieuws is, anders moet, beter, sneller. Als managers zien we het belang, en steunen we de noodzakelijke veranderingen, maar hoe

zorg je er voor dat tijdens de verbouwing de operatie doorgaat? Hoe hou je overzicht en inzicht? En, hoe zorg je ervoor dat de druk niet te hoog wordt, zodat je medewerkers betrokken en gemotiveerd blijven?

Het leven van project en lijnmanagers is er niet eenvoudiger opgeworden, en de verwachtingen zijn hoog. Software Process Improvement en Kwaliteits Management hebben oplossingen. Op deze conferentie zoeken we presentaties uit de praktijk van alledag, SPI en kwaliteit toegepast om met veranderingen om te gaan, en beter en sneller te werken. We nodigen je uit om inspirerende en interessante ervaringen met vakgenoten te delen. Ook doen we een oproep voor onconventionele presentaties en nieuwe ervaringen, die anderen aan het denken zetten en tot creatieve prestaties kunnen brengen.

Mogelijke onderwerpen binnen dit thema:

- Lerende organisaties, retrospectives, managen en verbeteren van kennis/vaardigheden
- Verander management, leiding geven, coaching
- Regie, Out-, near- en insourcing, fusies, Co-development, multi-site
- Standaarden, Certificering, auditing (SOX, BASEL2, SEPA, MUFFIT, COBIT, etc)
- Werven, motiveren, behouden van professionals
- Modellen, toepassing, multimodel (Agile, lean, Six Sigma, CMMI, People CMM)
- Teamworking, PSP, TSP

Indien je een presentatie wilt geven op de conferentie, vul dan het registratieformulier van de Call for papers op de website in, en stuur het naar info@spiderconferentie.nl. Als je presentatie geprogrammeerd wordt, kun je gratis deelnemen aan de conferentie.

Denk aan de deadline: Insturen voor 21 maart!

Namens de Programmaraad
Ben Linders, voorzitter Stichting SPIDER

■ CMMI roadmaps worden gepubliceerd bij het SEI

Roadmaps zijn verschillende routes die organisaties bij de implementatie van CMMI kunnen kiezen. Deze routes zijn afhankelijk van de op te lossen problemen of te bereiken doelen. Er worden vijf roadmaps onderkent:

- Project
- Product
- Product Integration
- Process
- Measurement

Eind 2006 is er, mede naar aanleiding van een publicatie in een IT-vakblad en presentatie op een conferentie, een workshop georganiseerd om het idee van roadmaps uit te werken. Tijdens de workshop hebben 35 CMMI specialisten, allemaal SPIDER-leden, de verschillende roadmaps uitgewerkt in concrete procesgebieden. Vervolgens hebben André Heijstek, Ben Linders, Rini van Solingen en ondergetekende de uitkomsten van de workshop verwerkt in het SEI template voor een technical note. Daarna is de technical note ter review aangeboden aan de deelnemers aan de workshop. De reviewopmerkingen zijn verwerkt in de technical note.

De technical note is in juni 2007 aangeboden aan het SEI. Op 30 november kregen de auteurs van de technical note een mail van Mike Konrad van het SEI met onder andere de volgende tekst:

"Bottom line: I think we should publish this - it should prove to be of practical use to many organizations and help broaden pursuit of process improvement and CMMI. We thank you and your co-authors for the very fine effort!"

Een stukje tekst wat voor zich spreekt. Naast deze bottom line had Mike enkele opmerkingen die we verwerkt hebben. De zogenaamde technical writers nemen het document nu onder handen om het helemaal rijp voor publicatie te maken.

Het gaat dus echt gebeuren, de technical note gaat gepubliceerd worden door het SEI! Daarmee wordt de technical note de eerste officiële Nederlandse bijdrage aan het CMMI. Zodra de technical note officieel gepubliceerd is brengen we jullie op de hoogte, daarnaast denken we over de manier waarop we de inhoud van de technical note binnen SPIDER kenbaar gaan maken.

Jan Jaap Cannegieter
Werkgroepleider Roadmaps

■ Laatste bijeenkomst van de metriecken werkgroep

In november vorig jaar was er een bijzondere meeting van de metriecken werkgroep. Bijzonder, niet alleen vanwege de onderwerpen maar ook omdat het de laatste was.

Na jarenlang kennis en inzichten te delen, bespreken van methodieken rondom het opzetten van meetprogramma's en het delen van praktijkkennis is het tijd om de werkgroep af te ronden.

De opkomst was de laatste tijd wat beperkt, wat een duidelijke indicator is dat we blijkbaar met elkaar gedeeld hebben wat er te delen valt. Tijd voor nieuwe initiatieven, nieuwe werkgroepen en nieuwe onderwerpen om ons op te storten.

Benjamin Jurg
Werkgroepleider Metriecken

**11e SPIDER conferentie
29 september 2008
Ede**

■ Deelname in SPIder

Indien u actief wilt participeren in SPIder en de Koerier in de toekomst wilt ontvangen, kunt u zich aanmelden als deelnemer in SPIder bij:

Secretariaat Stichting SPIder

p/a Cantrijn Secretariaten
Postbus 2047, 4200 BA GORINCHEM
Tel: 0183 - 62 00 66, fax: 0183 - 62 16 01
E-mail: info@st-SPIder.nl

Aanmelding kan ook via het aanmeldingsformulier op de website van SPIder: www.st-SPIder.nl.

■ Nieuwsberichten & evenementenkalender

De evenementenkalender bevat een overzicht van internationale conferenties op het gebied van SPI, metriecken en softwareproductkwaliteit. Daarnaast zijn de activiteiten van SPIder opgenomen.

Ook nationale evenementen op het gebied van softwareproduct- en procesverbetering kunnen in deze evenementenkalender worden opgenomen. Via de SPIder Koerier kan een organisator van SPI gerelateerde evenementen een selecte groep van geïnteresseerden bereiken. Voor commerciële evenementen zoals conferenties, workshops, lezingen en andersoortige bijeenkomsten vraagt de redactie een kleine bijdrage in de kosten.

Ⓢ = SPIder event

✓ = korting voor SPIder donateurs

■ Colofon

De SPIder redactie bestaat uit:
Cees Michielsen en Cantrijn Secretariaten.

Voor reacties en vragen m.b.t. de **SPIder Koerier** kunt u zich wenden tot:
Redactie SPIder Koerier
E-mail: koerier@st-SPIder.nl

Indien u in de toekomst een herinneringsbericht wilt ontvangen over de datum van kopijsluiting, stuur dan een e-mail "opname SPIder copylijst" naar Koerier@st-SPIder.nl.

Informatie over SPIder is te vinden op de website: www.st-SPIder.nl.

Voor reacties en bijdragen op de **SPIder website** kunt u zich richten tot:
Redactie SPIder web, Niels Malotaux
E-mail: niels@malotaux.nl

Deze koerier kwam tot stand met medewerking van:
- [ITIB](http://www.itib.net) (www.itib.net)
- Cantrijn Secretariaten

2008

26 feb.	Testnet: Thema-avond - Werkgroep Outsourcing en Metriecken www.testnet.org/Produktie/Evenementen/ThemaAvond.html
Ⓢ 6 maart	Plenaire sessie: Multi-site ontwikkelen
6 maart	ASL BiSL Foundation: Themabijeenkomst 'Regie en outsourcing met BiSL'
Ⓢ 11 maart	Werkgroep Invoeringsstrategieën Contact: Andre Heijstek
12 maart	Software VOC netwerk: Het gebruik van open source componenten bij software productie www.ictoffice.nl
17 maart	SEPG North America
27 maart	Testnet: Thema-avond - Testontwerptechnieken www.testnet.org/Produktie/Evenementen/ThemaAvond.html
8 april	IPMA Nederland: de Project Management Parade www.projectmanagementparade.nl
10 april	NESMA: Prestaties Meten en Begroten www.nesma.nl
17 april	Nesma voorjaarsconferentie www.nesma.nl
21 april	Testnet: Thema-avond - EuroSTAR mini-event www.testnet.org/Produktie/Evenementen/ThemaAvond.html
24 april	LaQuSo: VVSS symposium, Software Quality in Outsourcing
Ⓢ April/mei	Q conferentie: informatie volgt
8 mei	ICTDelta, het innovatiecongres www.ictdelta.nu
Ⓢ 13 mei	Werkgroep Invoeringsstrategieën Contact: Andre Heijstek
Ⓢ 22 mei	Plenaire sessie: Requirements
20 mei	Testnet: Thema-avond - Testomgevingen www.testnet.org/Produktie/Evenementen/ThemaAvond.html
10-13 juni	SEPG Europe Conference
Ⓢ 19 juni	Plenaire sessie: 'Testen / Test Improvement'
15-19 juni	INCOSE International Symposium 2008 - Utrecht www.incose.org/symp2008
19 juni	Bits&Chips, Hardware conference www.hardwareconference.nl
Ⓢ 24 juni	Werkgroep Invoeringsstrategieën Contact: Andre Heijstek
30 juni	Testnet: Thema-avond - Tools voor testen www.testnet.org/Produktie/Evenementen/ThemaAvond.html
Ⓢ Sept.	Plenaire sessie: 'Enterprise Architecture'
3 sept.	EuroSPI Conference www.eurospi.net ✓
Ⓢ 16 sept.	Werkgroep Invoeringsstrategieën Contact: Andre Heijstek
16 sept.	Testnet: Najaarsevenement Thema TESTEN: DE WAARHEID! www.testnet.org/Produktie/Evenementen/ThemaAvond.html
Ⓢ 29 sept.	11e SPIder conferentie www.spiderconferentie.nl ✓
8 okt.	Bits & Chips Embedded systemen www.bits-chips.nl
23 okt.	Testnet: Thema-avond - Wet en regelgeving www.testnet.org/Produktie/Evenementen/ThemaAvond.html
Ⓢ 4 nov.	Werkgroep Invoeringsstrategieën Contact: Andre Heijstek
19 nov.	Testnet: Thema-avond - Werkgroepen www.testnet.org/Produktie/Evenementen/ThemaAvond.html
16 dec.	Testnet: Thema-avond - Op locatie www.testnet.org/Produktie/Evenementen/ThemaAvond.html